

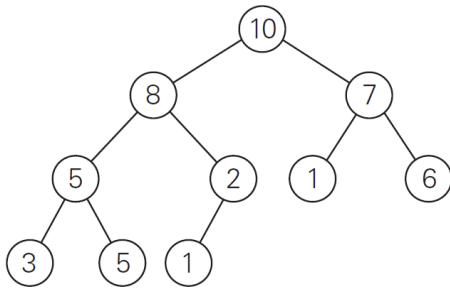
Name: _____

Date: _____

Point values are assigned for each question.

Points earned: ____ / 100

1. Consider the algorithm on page 148 in the textbook for binary reflected Gray codes. What change(s) would you make so that it generates the binary numbers **in order** for a given length n ? Your algorithm must be recursive and keep the same structure as the one in the textbook. Describe only the change(s). (10 points)
2. Show the steps to multiply 72×93 with Russian peasant multiplication, as seen in Figure 4.11b on page 154 in the textbook. (10 points)
3. Suppose you use the `LomutoPartition()` function on page 159 in the textbook in your implementation of quicksort. (10 points, 5 points each)
 - a. Describe the types of input that cause quicksort to perform its worst-case running time.
 - b. What is that running time?
4. Compute 2205×1132 by applying the divide-and-conquer algorithm outlined in the text. Repeat the process until the numbers being multiplied are each 1 digit. For each multiplication, show the values of c_2 , c_1 , and c_0 . Do not skip steps. (10 points)
5. Draw the binary search tree after inserting the following keys: 24 18 67 68 69 25 19 20 11 93 (10 points)
6. Consider the following binary tree. (16 points, 2 points each)



- a) Traverse the tree preorder.
 - b) Traverse the tree inorder.
 - c) Traverse the tree postorder.
 - d) How many internal nodes are there?
 - e) How many leaves are there?
 - f) What is the maximum width of the tree?
 - g) What is the height of the tree?
 - h) What is the diameter of the tree?
7. Use the Master Theorem to give tight asymptotic bounds for the following recurrences. (25 points, 5 points each)

- a) $T(n) = 2T(n/4) + 1$
- b) $T(n) = 2T(n/4) + \sqrt{n}$
- c) $T(n) = 2T(n/4) + n$
- d) $T(n) = 2T(n/4) + n^2$
- e) $T(n) = 2T(n/4) + n^3$

8. Consider the following function. (9 points)

```
int function(int n) {
    if (n <= 1) {
        return 0;
    }
    int temp = 0;
    for (int i = 1; i <= 6; ++i) {
        temp += function(n / 3);
    }
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j * j <= n; ++j) {
            ++temp;
        }
    }
    return temp;
}
```

- a) Write an expression for the runtime $T(n)$ for the function. (4 points)
- b) Use the Master Theorem to give a tight asymptotic bound. Simplify your answer as much as possible. (5 points)